AMENDMENTS TO THE SPECIFICATION

Please amend the specification, as follows:

Replace paragraph [0005] with the following amended paragraph [0005]:

A zero-stopping incrementer, using many static logic gates, is discussed in U.S. Patent No. 5,635,858. The zero-stopping incrementer determines whether the input operand is an even number or odd number. If the input operand is an even number, the zero-stopping incrementer changes a least significant bit (LSB) to a binary “1”. For an odd number, the zero-stopping incrementer searches for the first binary “0” beginning with the LSB, changes that binary “0” to a binary “1” and all preceding binary “1s” [[“0s”]] into binary “0s” [[“1s”]]. However, the zero-stopping incrementer occupies a large area of the microprocessor chip due to use of many static logic gates.

Replace paragraph [0010] with the following amended paragraph [0010]:

In an exemplary embodiment, the increment is performed in accordance with the following Boolean logic expressions[[,]]:

IF IN<0> and IN<1> = “1”,

(IN+1)<3:2> = IN_C<3:2>,

(IN+1)<1:0> = “00”,

IF IN<0> or [[and]] IN<1> = “0”,

(IN+1)<3:2> = IN<3:2>,

(IN+1)<1:0> = IN_C<1:0>,

where IN represents the operand, IN+1 represents the increment value, IN_C represents a newly defined increment value, "0" represents the first logic state, and "1" represents the second logic state (Boolean expression 1), and

IN_C<0> = ~IN<0>,

IN_C<1> = IN<0> OR IN<1>,

IN_C<2> = ~IN[[c]]<2>,

IN_C<3> = IN<2> OR IN<3>,

where IN represents the operand and IN_C represents a newly defined increment value (Boolean expression 2). The tilde symbol "~" represents the logical compliment and the "OR" symbol represents exclusive-or.

Replace paragraph [0011] with the following amended paragraph [0011]:

In an exemplary embodiment, the logical combination is performed in accordance with the following Boolean logic expression[[,]]:

IF ZD = "0" and CA = "0", IO = "0000",

IF ZD = "1" and CA = "0", IO = IN+1,

IF CA = "1" where ZD is any value, IO = IN,

where ZD represents the first logic state inclusion information for each 4-bit group, CA represents the flag information for each 4-bit group, IO represents the whole increment value for each 4-bit group, IN represents the operand, IN+1 represents the increment value, "0" represents the first logic sate, and "1" represents the second logic state (Boolean expression 3).

Replace paragraph [0025] with the following amended paragraph [0025]:

The operand IN is generally composed of 32 bits or 64 bits in microprocessor calculations, but may be composed of any number of 2^N bits where N is zero or a positive [[an]] integer. The incrementer according to exemplary embodiments of the present invention can process any number of bits[[,]]. However, ~~however~~ for the following examples, the operand IN is assumed to be composed of 32 bits.

Replace paragraph [0026] with the following amended paragraph [0026]:

If the 32-bit operand IN is divided into eight 4-bit groups, the first logic state inclusion information ZD for each 4-bit group, output from the 4-bit zero detection unit 110, is composed of 1 bit, and thus, a total of 8 bits of first logic state inclusion information ZD is generated. For a 4-bit group with a logic "0", the first logic state inclusion information ZD is output as a logic "1". For a 4-bit group with only [[a]] logic "1s", the first logic state inclusion information ZD is output as a logic "0". In other embodiments of the present invention, the operand IN can be divided into different b-bit groups. For example, if the operand IN is divided into six 6-bit groups, the first logic state inclusion information ZD can be 1 bit[[,]] and, thus, a total of 6 bits of first logic state inclusion information ZD is generated. Other embodiments in accordance with the present invention can used various bit sizes for the operand IN and the first logic state inclusion information ZD.

Replace paragraph [0032] with the following amended paragraph [0032]:

The increment value ADD is output in accordance with the following Boolean logic expressions[[.]]:

IF $IN<0>$ and $IN<1> =$ "1",

$(IN+1)<3:2> = IN_C<3:2>$,

$(IN+1)<1:0> =$ "00",

IF $IN<0>$ or [[and]] $IN<1> =$ "0",

$(IN+1)<3:2> = IN<3:2>$,

$(IN+1)<1:0> = IN_C<1:0>$, ... (1),

where IN represents the operand, IN+1 represents the increment value ADD, IN_C represents a newly defined increment value, "0" represents the first logic state, and "1" represents the second logic state[[.]], and

$IN_C<0> = \sim IN<0>$,

$IN_C<1> = IN<0>$ OR $IN<1>$,

$IN_C<2> = \sim IN$[[c]]$<2>$,

$IN_C<3> = IN<2>$ OR $IN<3>$, ... (2),

where IN represents the operand, [[and]] IN_C represents a newly defined increment value (Boolean expression 2). The tilde symbol "~" represents the logical compliment and the "OR" symbol represents exclusive-or.

Replace paragraph [0038] with the following amended paragraph [0038]:

Referring to FIG. 2, when a clock signal CLK is inactivated, i.e., the first logic state, a PMOSFET P2 and an input end [[nodes]] of an inverter INV21 ~~and a PMOSFET P2~~ are precharged to a precharging voltage VDD by a PMOSFET P1. The inverter INV21 and the PMOSFET P2 function as output buffers. The inverter INV21 inverts the precharging voltage VDD, and the PMOSFET P2 supplies the precharging voltage VDD to an input end of the inverter INV21 in response to the output of the inverter INV21. When the clock signal CLK is activated, i.e., the second logic state, and all pairs of NMOSFETs, e.g., N1 and N2, and N3 and N4, etc., connected in series between ~~the output nodes of~~ the PMOSFET P2 (and the input end of the inverter INV21) and a ground voltage, are activated, the output of the inverter INV21 is converted into the second logic state. NIN<0> through NIN<3> are 4-bit inverted signals, belonging to a 4-bit group among eight 4-bit groups constituting the 32-bit operand IN. Each 4-bit group needs the 4-bit zero detection unit 110 of FIG. 2. In alternative embodiments of the present invention, operand IN can be of various bit sizes, and the group size can vary in relation to operand IN.

Replace paragraph [0041] with the following amended paragraph [0041]:

Referring to FIGS. 3A and 3B, a PMOSFET P32 and an input end ~~output nodes~~ of an inverter INV31 ~~and a PMOSFET P32~~, i.e., a node ZZCA, are precharged to the precharging voltage VDD by a circuit 320 when the clock signal CLK input through a clock buffer 310 is inactivated, i.e., the first logic state. The inverter INV31 and the PMOSFET P32 function as

output buffers. The inverter INV31 inverts the precharging voltage VDD. The PMOSFET P32 supplies the precharging voltage VDD to the input end of the inverter INV31, i.e., the node ZZCA, in response to the output of the inverter INV31. The output of the inverter INV31 is converted into the second logic state when the clock signal is activated, i.e., the second logic state, and a plurality of NMOSFETs 330, connected in series between ~~the input terminal of~~ the PMOSFET P32 (and the input end of the inverter INV31), i.e., the node ZZCA, and a ground voltage, are activated. ZD<0> through ZD<6> represent the first logic state inclusion information ZD for each 4-bit group. Since the flag information CA<7> is a logic "0" if the first 4-bit group starting from the most significant bit (MSB) has a logic "0", ZD<7> is not needed for the increment output unit 140 of FIG. 5 to generate the flag information CA for each 4-bit group. Besides, since CA<0> is a logic "0" at all times, CA<0> is not needed either. In alternative embodiments of the present invention CA<0> is used and/or ZD<7> is used.

Replace paragraph [0044] with the following amended paragraph [0044]:

Referring to FIGS. 4A and 4B, the 4-bit increment unit 130 of FIG. 1 includes increment circuits 420 through 450 for 4 bits. When the clock signal CLK input through a clock buffer 410 is inactivated, i.e., the first logic state, ~~respective input nodes of~~ each paired PMOSFET [[of]] and input end of an inverter ~~and a PMOSEFT~~, e.g., an inverter INV41 and a PMOSFET P12, are precharged to the precharging voltage VDD by a circuit, e.g., PMOSFET P11. The paired inverters and [[the]] PMOSFETs ~~PMOSEFT~~ function as output buffers. Each inverter INV41, INV42, INV43, and [[or]] INV44 inverts the precharging voltage VDD, and each PMOSFET P12, P14, P16, and P18 supplies the precharging voltage VDD to the input end of a paired

inverter ~~each PMOSFET~~ according to the output of the [[a]] paired inverter. The output of the paired inverter is converted into the second logic state when the clock signal CLK is activated, i.e., the second logic state, and NMOSFETs N11, N12, N13, N14, N16, N17, N20, and N21, connected in series between ~~the input ends of~~ the PMOSFET (and the input end of the paired inverter) and a ground voltage, are activated. IN<0> through IN<3> represent 4-bit signals belonging to one 4-bit group of eight 4-bit groups of the 32-bit operand IN. NIN<0> through NIN<2> represent inverted signals of IN<0> through IN<3>. Each 4-bit group can use the 4-bit increment unit 130 of FIGS. 4A and 4B.

Replace paragraph [0048] with the following amended paragraph [0048]:

Referring to FIG. 5, when the clock signal CLK input through an output buffer 510 is inactivated, i.e., the first logic state, PMOSFET P52 and an input end ~~the input nodes~~ of an inverter INV51 ~~and a PMOSFET P52~~ are precharged to the precharging voltage VDD by a circuit, e.g., PMOSFET P51. The inverter INV51 and the PMOSFET P52 function as output buffers. The inverter INV51 inverts the precharging voltage VDD. The PMOSFET P52 supplies the precharging voltage VDD to the input end of the inverter INV51 in response to the output of the inverter INV51. The output of the inverter INV51 is inverted to the second logic state when the clock signal CLK is activated, i.e., the second logic state, and NMOSFETs N51 through N54, connected in series between ~~output ends of~~ the PMOSFET P52 (and the input end of the inverter INV51) and the ground voltage, are activated. NCA represents an inverting signal of the flag information CA. The increment output unit 140 of FIGs. 1 and 5 are needed for each bit of the

32-bit operand IN. NCA, CA, and ZD can be composed of 4 bits, respectively. IN and ADD correspond with each other for each bit ~~each bit~~ and are each composed of 32 bits.